# Programación Evolutiva
## Algoritmo Evolutivo. Optimización Combinacional
## Práctica 2

**Miguel Márquez Altuna**
**Fco Javier Álvarez Obeso**

## Índice

1. Introducción. Algoritmo evolutivo..... 3
2. Métodos propios..... 3
    2.1. Método de selección. *High-Low*..... 3
    2.2. Método de cruce. *Antienemies*..... 4
3. Resultados más significativos..... 7
    3.1. Ensayo 1. Comparativa entre métodos de selección..... *7*
    3.2. Ensayo 2. Métodos de cruce..... *10*
        3.2.1. Comparativa cruce OX, PMX y Ordinal..... *10*
        3.2.2. Cruce *Antienemies* ..... *13*
    3.3. Ensayo 3. Mutación heurística..... *15*
    3.4. Ensayo 4. Función de evaluación..... *16*
4. Conclusiones..... *17*

# 1. Introducción. Algoritmo evolutivo

*El objetivo de esta práctica es implementar, utilizando en lo posible el esquema de la práctica anterior, un algoritmo evolutivo para optimizar la asignación de alumnos a grupos en un sistema automático de formación de grupos.*

Para llevar a cabo la implementación del algoritmo evolutivo nos hemos basado en la implementación que ya se había realizado del algoritmo genético realizando una generalización adicional, para que el contenido de cada gen dentro de cada cromosoma pudiera contener un tipo T parametrizable cualquiera.

También, ante el crecimiento de operadores de cruce, de mutación así como de selección se generalizaron las jerarquías para que también fueran parametrizables conforme a ese tipo T, que será el mismo que contengan los genes de los cromosomas.

Cada uno de estos genes contiene una lista de tipos genéricos que representará a un alumno, más concretamente su posición en cierta estructura de datos. Los cromosomas contienen conjuntos de estos genes, que representarán las distintas distribuciones, considerando aquellos genes en posiciones consecutivas como miembros de un mismo grupo. Será tarea de las técnicas de cruce y mutación que no se llegue a conflictos ni aparezcan genes replicados dentro de los cromosomas.

# 2. Métodos propios

## 2.1. Método de selección. *High-Low*

La idea sobre la que se basa este método es la de tratar de conservar cierta homogeneidad en la población, para ello tratará de añadir a la población futura algunos de los mejores con alguno de los peores. Esta distinción se hará mediante un parámetro d.

Los pasos que realiza son los siguientes:

1. Ordena la población según su adaptación, de forma que podamos distinguir fácilmente los mejores.

2. La divide en dos conjuntos h y l. Esta distinción se realiza en base a un parámetro que indicará qué porcentaje de individuos consideramos peores y definirá la partición de la población.

3. Se toma al azar un individuo de uno u otro conjunto hasta completar la población intermedia.

Los resultados obtenidos no son del todo positivos para ciertos valores del parámetro. Por lo general, los mejores resultados se obtienen para los valores extremos del parámetro (es decir,

alrededor del 0 y del 1); o lo que es lo mismo, cuando uno de los conjuntos es muy grande y el otro muy pequeño.

## 2.2. Método de cruce. *Antienemies*

Considerados 2 cromosomas cualesquiera, este método trata de que la asignación de grupos resultante en los hijos tenga el menor número posible de alumnos que sean incompatibles, es decir el número de alumnos que estén en un grupo con otros que no quieren. Para ello a la hora de generar un hijo se realizan los siguientes pasos:

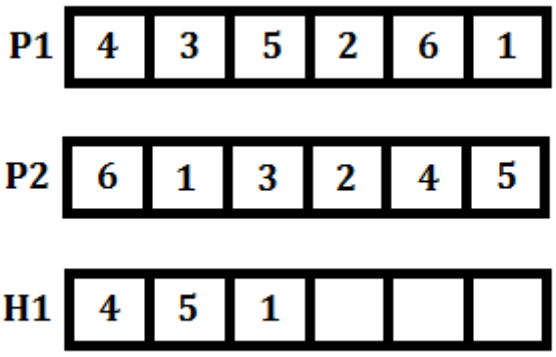
1. Se considera alternativamente un alumno (de ahora en adelante referidos como "genes") de un padre y del otro. Basta realizar recorridos alternativos de los padres para obtener varios hijos es decir, orden creciente, orden decreciente, al azar *n* posiciones...
2. A cada gen le asignaremos un grupo dentro del hijo que podrá ser:
   - Uno con miembros existentes si estos no presentan incompatibilidades con él.
   - Si el alumno presenta incompatibilidades con todos los grupos con miembros se toma el primero libre.
   - Si además estuvieran todos ocupados se busca el grupo restante que presente menor número de incompatibilidades.

Esto puede generar conflictos en el hijo que se solucionan ignorando aquellos alumnos que ya están presentes en algún grupo del nuevo hijo.

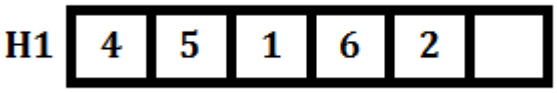
Supongamos las siguientes incompatibilidades:

| Alumno | Alumnos incompatibles |
|---|---|
| 1 | 2, 4 |
| 2 | 4 |
| 3 | 2, 5 |
| 4 | 1, 2 |
| 5 | - |
| 6 | 4, 5 |

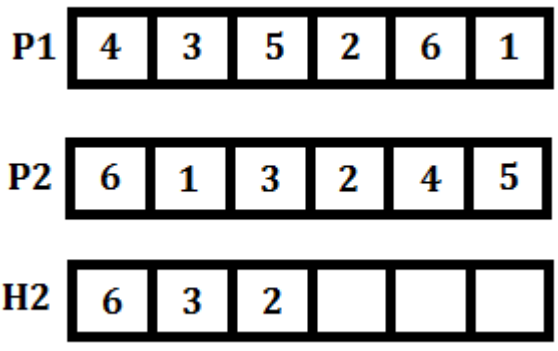
Además consideramos los padres p1 y p2, para formar grupos de 2 se obtiene:



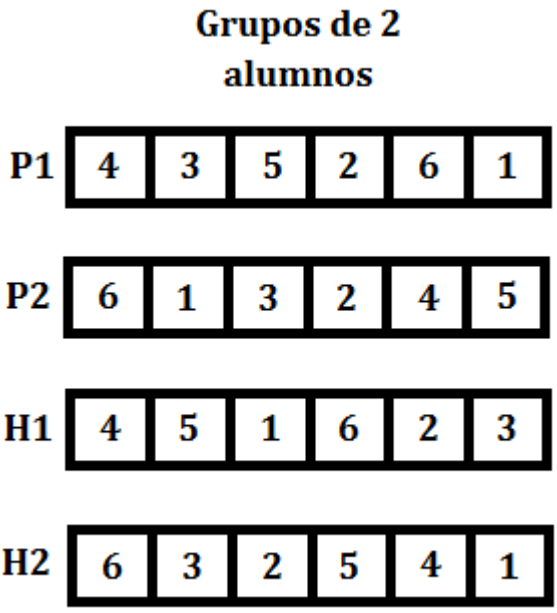

En esta solución parcial vemos que el primer grupo se ha completado, y el segundo ya tiene asignado su primer miembro. Para ello hemos recorrido alternativamente P1 y P2 (4, 1, 5, 2, 6, 5, 6...) y hemos aplicado el paso 2 explicado previamente. Como 1 presenta incompatibilidad con 2 le hemos asignado el siguiente grupo libre. 5 no presenta incompatibilidad alguna con el primer grupo y por tanto se le asigna.



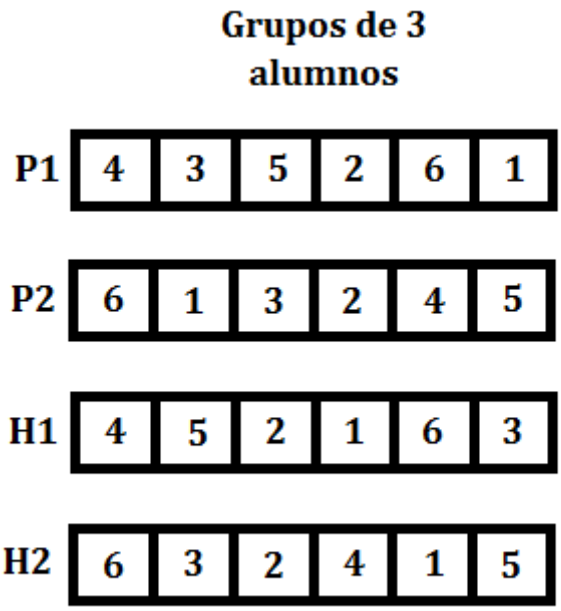

A continuación se asigna 2 al tercer grupo puesto que presenta incompatibilidad con 1, y 6 al segundo por la misma razón.
El segundo hijo se genera con la misma dinámica recorriendo los padres de forma diferente, en este caso empezamos la secuencia por p2 en vez de p1 (6, 3, 3, 2...)




En este caso vemos como el alumno 3 repetido se ignora y se sigue la secuencia por dos. Hasta obtener el resultado final:




Otro ejemplo adicional, considerando los mismos padres e incompatibilidades, para formar grupos de 3 se obtiene:



# 3. Resultados más significativos

Aunque en la interfaz y la implementación están preparadas para incorporar de manera muy simple una nueva función de evaluación al problema, todos los ensayos consideran la función dada en el enunciado.

## 3.1. Ensayo 1. Comparativa entre métodos de selección

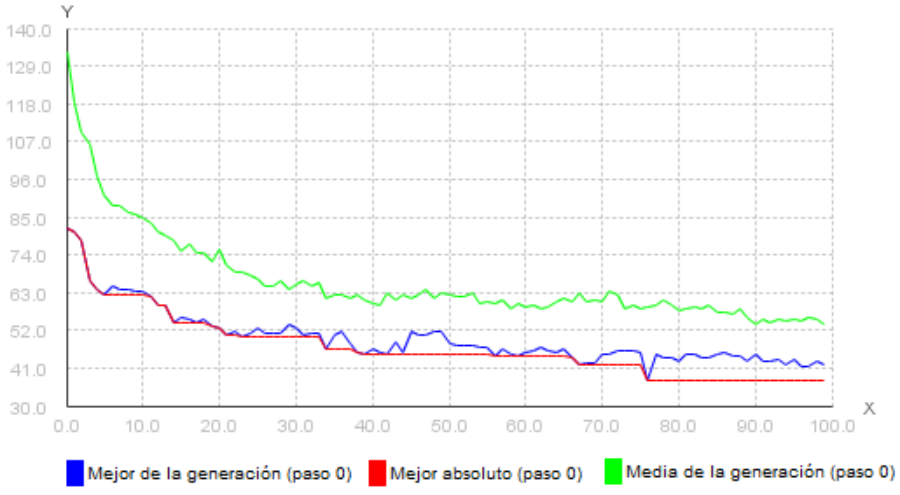
Vamos a comparar los diferentes métodos de selección tomando como referencia el fichero de pruebas 1, para ello consideramos la siguiente parametrización:

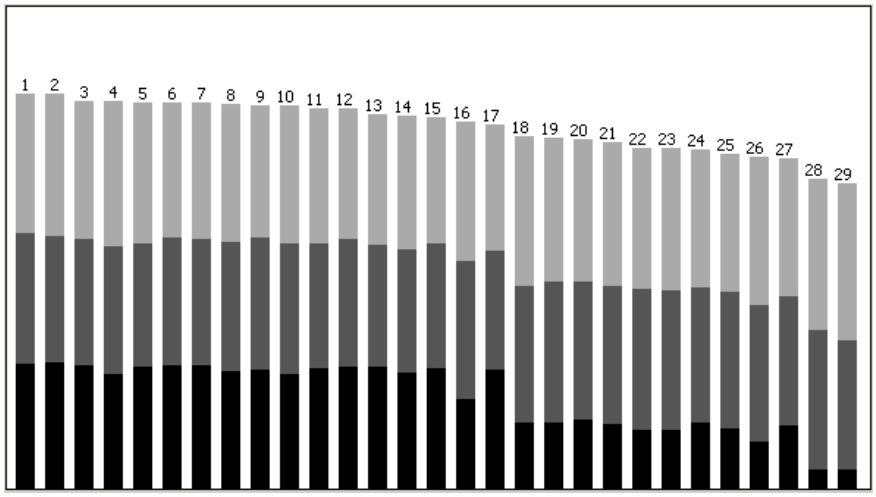
| Parámetro | Valor |
|---|---|
| Función de evaluación α | 0.5 |
| Tamaño de grupos m | 3 |

| Parámetro | Valor |
|---|---|
| Tam. Población | 100 |
| Prob. Cruce | 0.6 |
| Prob. Mutación | 1 |
| Número de generaciones | 100 |
| Cruce | PMX |
| Mutación | Intercambio |

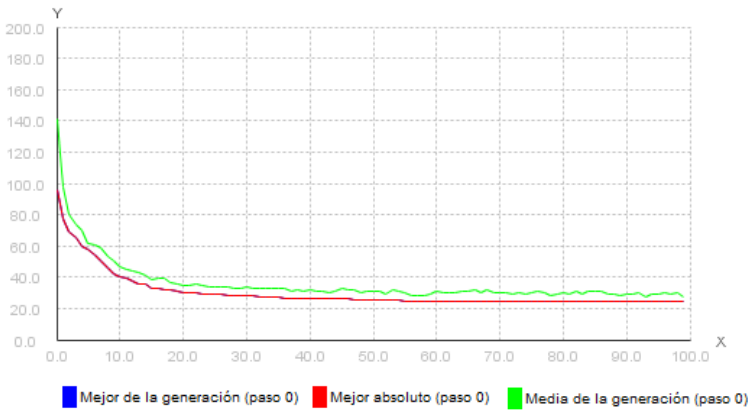
En base a lo anterior se obtienen los siguientes resultados:

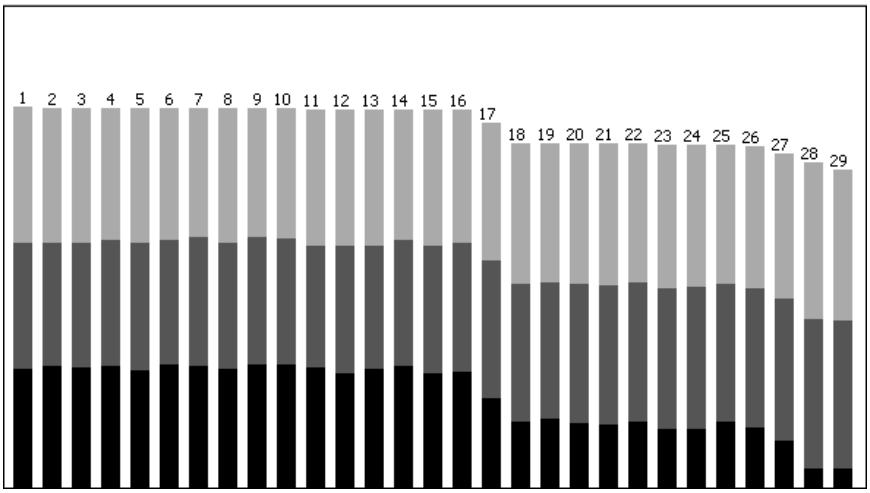
**Selección por ruleta**





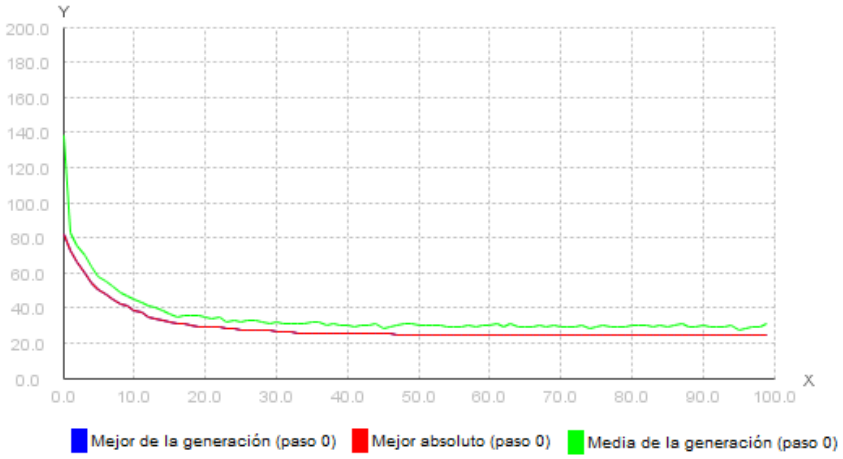

**Selección por Torneo**

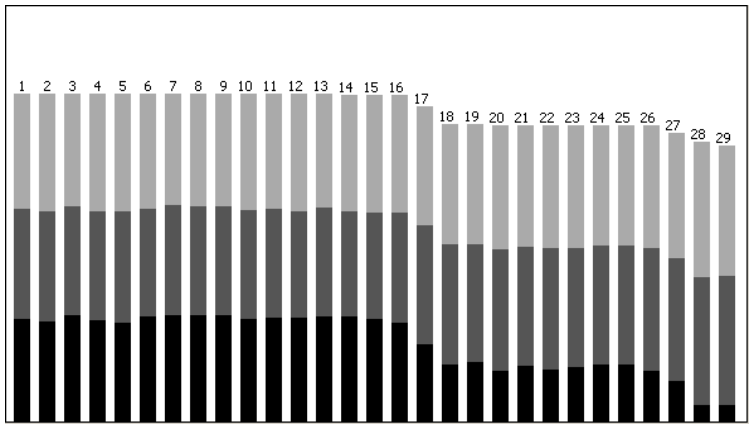



En general tanto para torneo determinista como probabilístico y cualesquiera sean sus parámetros no ofrecen una variación significativa de los resultados mostrados en las imagénes previas.

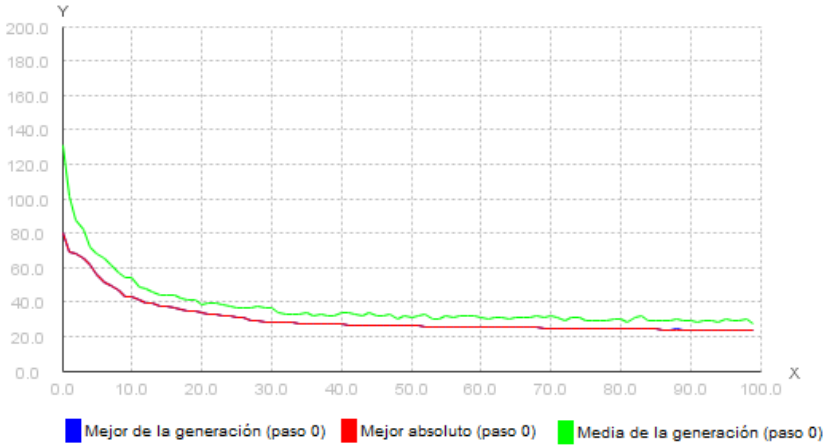
**Selección por ranking**

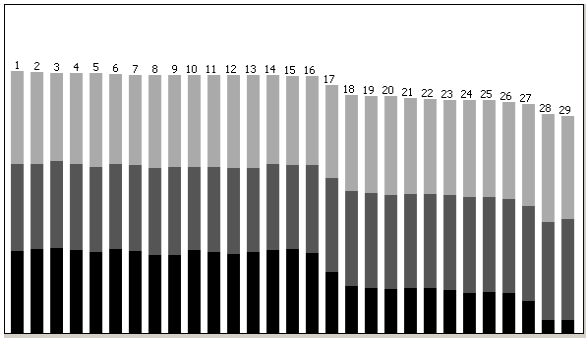



Con β = 1 aunque su variación en este caso no ofrece cambios significativos.

**Selección por *HighLow***






Con valores de parámetro entorno a 0.1.

En general, vemos que los métodos que ofrecen un mejor distribución conforme a la función especificada en el enunciado, son aquellos que están considerando valores "escalados" de la adaptación en base a su posición como ranking, que aquellos que tienen en cuenta los valores absolutos de adaptación. Esto ocurre en el caso de la ruleta donde vemos que al manejar los valores absolutos sin realizar ningún tipo de escalación obtenemos la peor distribución.
En el caso particular de *HighLow* para valores de su parámetro más bajos, ofrece unos mejores resultados, puesto que la élite seleccionada el conjunto h es mucho más pequeño y existen una mayor probabilidad de escoger un mejor individuo de h.

## 3.2. Ensayo 2. Métodos de cruce.

En esta sección se van a comentar aquellos comportamientos de los métodos de mutación cierta importancia.

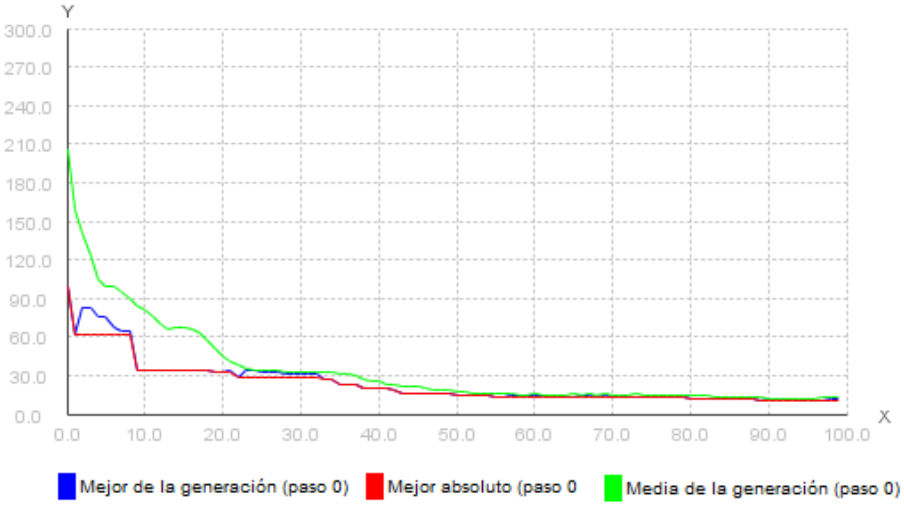
### 3.2.1. Comparativa cruce OX, PMX y Ordinal.

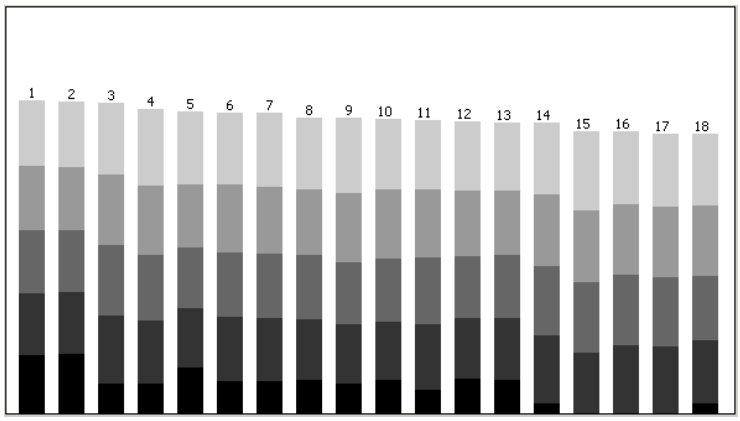
Considerado OX simple puesto que el alternativo produce resultados similares, PMX y ordinal vamos aplicar la misma parametrización sobre los 3 para ver qué resultados se obtienen considerando el conjunto de alumnos del fichero 1. Sean los siguientes valores:

| Parámetro | Valor |
|---|---|
| Función de evaluación α | 0.5 |
| Tamaño de grupos m | 5 |

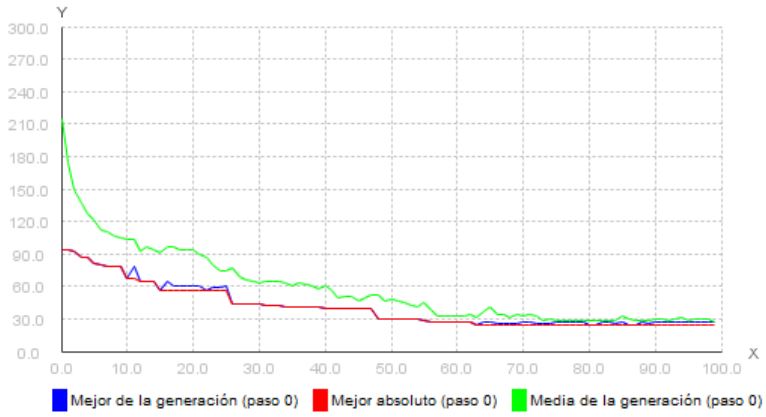
| Parámetro | Valor |
|---|---|
| Tam. Población | 100 |
| Prob. Cruce | 0.6 |
| Prob. Mutación | 0.05 |
| Número de generaciones | 100 |
| Selección | Ruleta |
| Mutación | Intercambio |

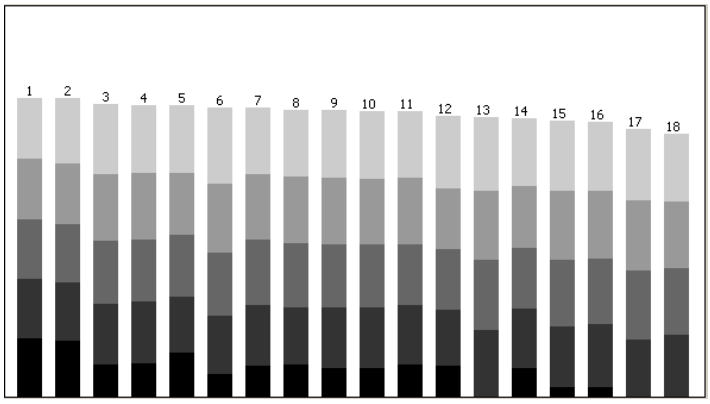
**Cruce PMX**



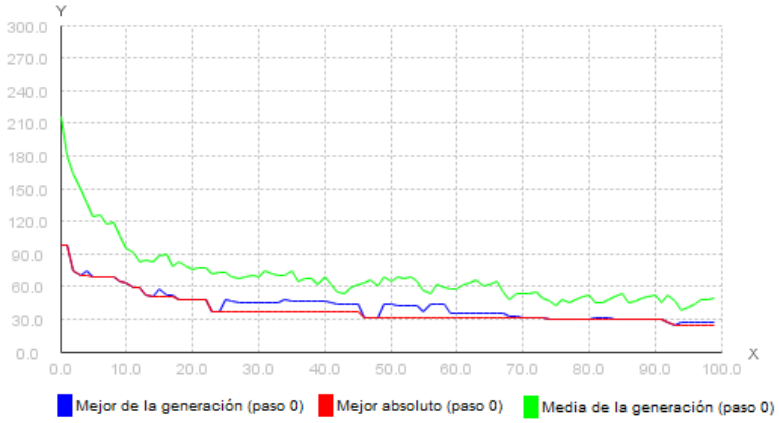



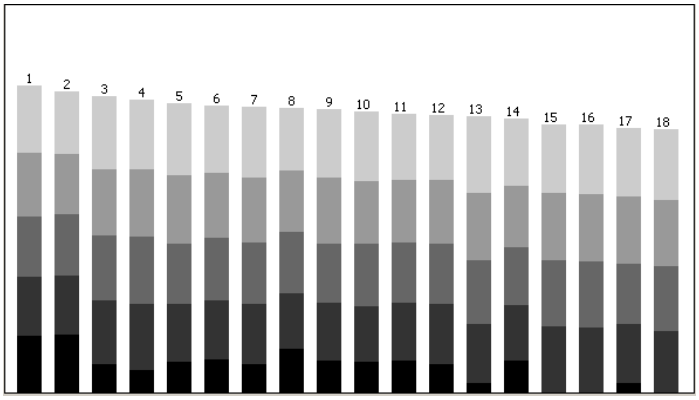

**Cruce OX**






**Cruce Ordinal**




Como se muestra en las gráficas PMX ofrece para este problema además de un mejor rendimiento, mejores resultados que el cruce ordinal o el cruce OX. De lo que se podría deducir que aquellas técnicas intentan conservar la distribución de los padres y no realizan una mezcla a partir de un punto elegido de forma arbitraria son más efectivas en este caso. Aún en el caso de OX sigue teniendo en cuenta...

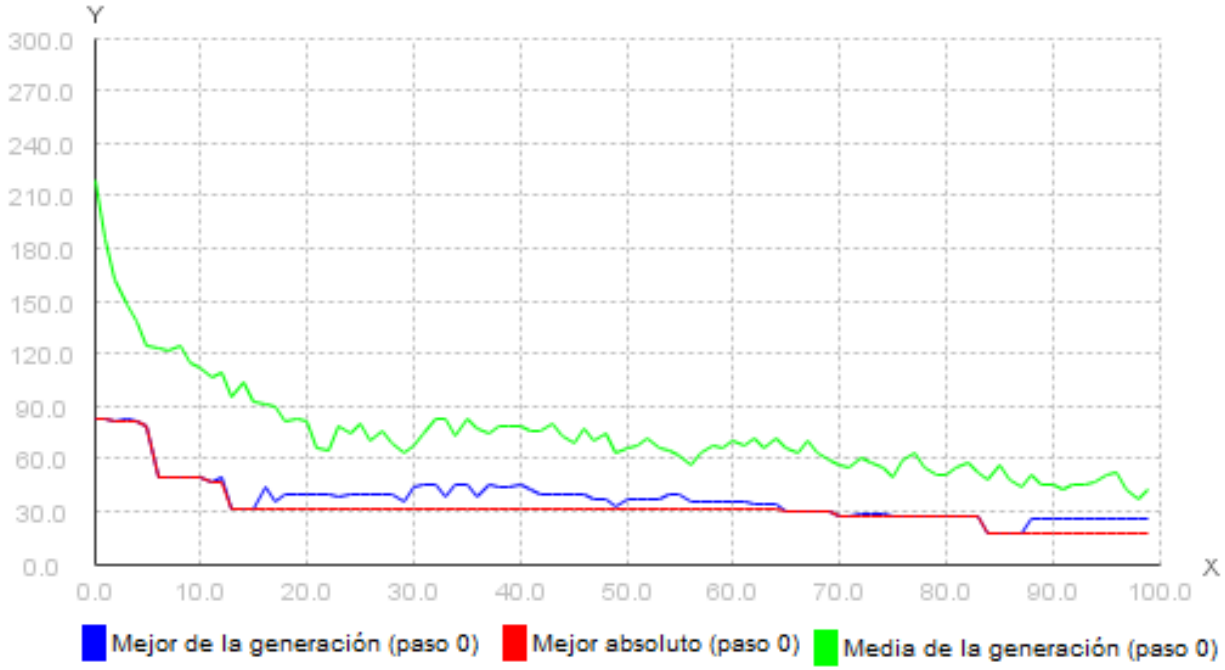
### 3.2.2. Cruce *Antienemies*

Este cruce como se comento al comienzo se basa en las incompatibilidades entre alumnos por ello en primer lugar vamos a comparar Antienemies con PMX para un fichero en el que existan incompatibilidades entre alumnos (Fichero 1) y luego haremos lo propio con otro en el que no haya incompatibilidades (Fichero 2) Considerada la siguiente parametrización:

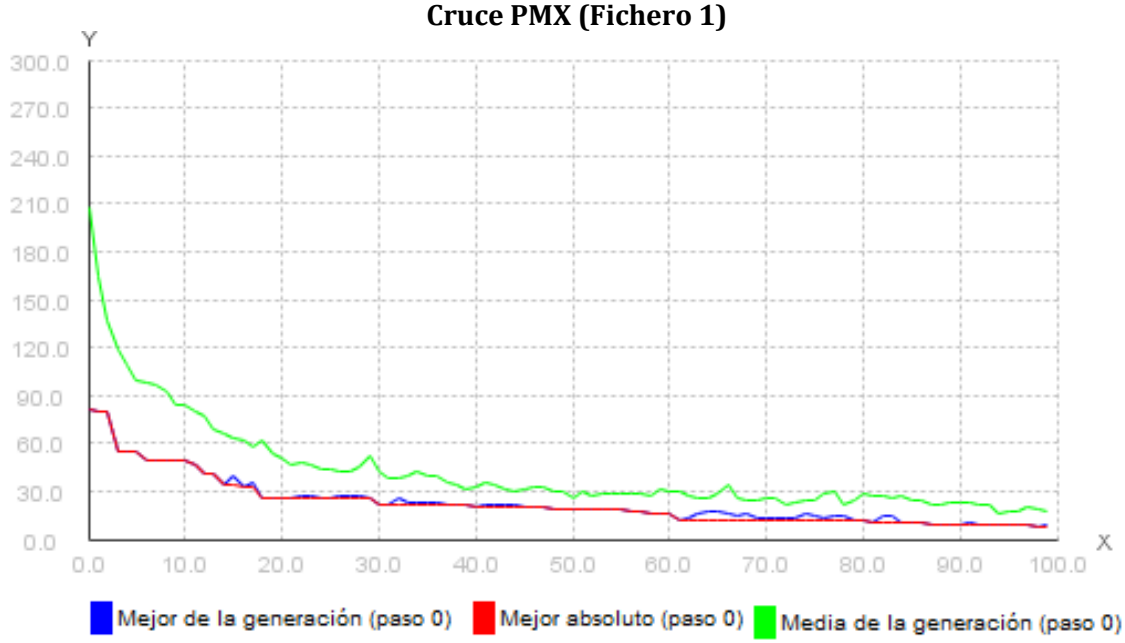
| Parámetro | Valor |
|---|---|
| Función de evaluación α | 0.5 |
| Tamaño de grupos m | 6 |

| Parámetro | Valor |
|---|---|
| Tam. Población | 100 |
| Prob. Cruce | 0.6 |
| Prob. Mutación | 0.2 |
| Número de generaciones | 100 |
| Selección | Ruleta |
| Mutación | Inversión |

**Cruce *Antienemies* (Fichero 1)**

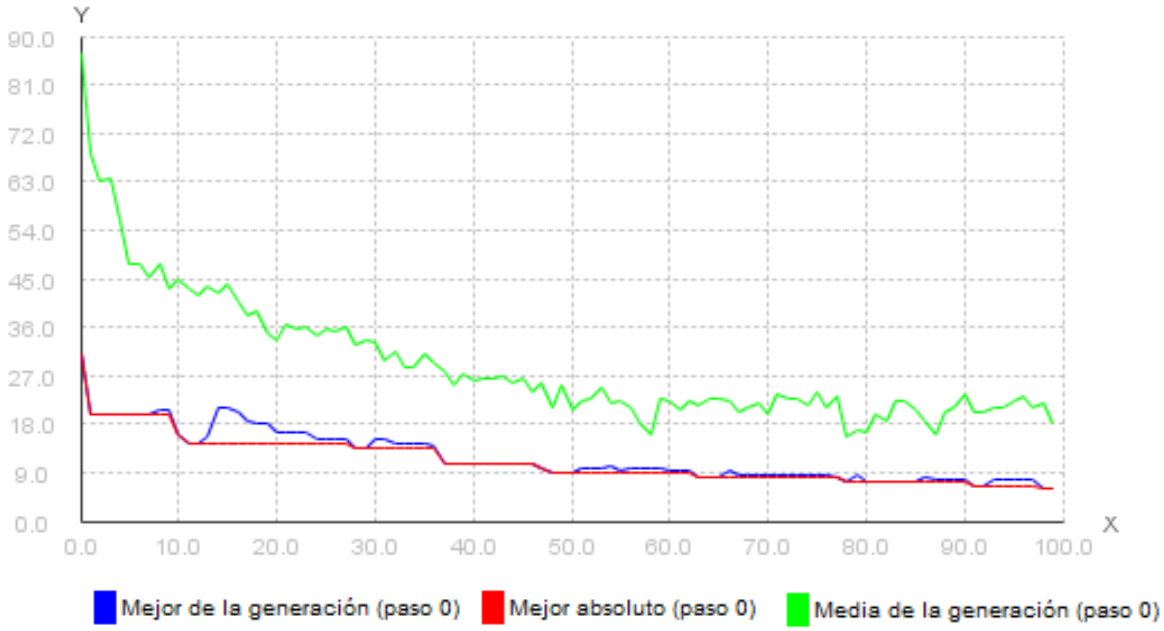


**Cruce PMX (Fichero 1)**

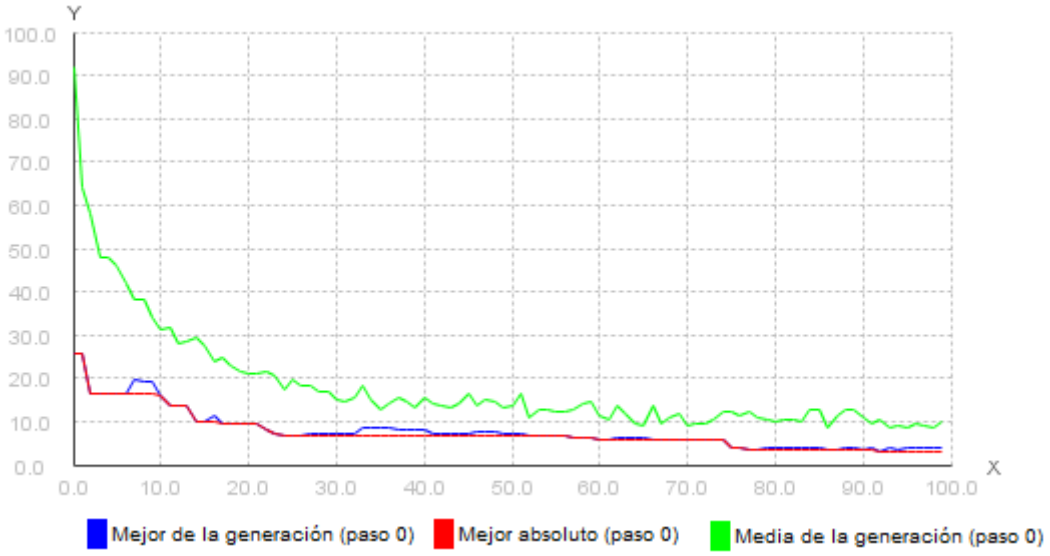



Como se puede ver en los resultados *antienemies* para un valor de su parámetro bajo entorno a 0.1 ofrece una mejor distribución similar a la del cruce PMX cuando entre los alumnos existen la penalización de que algunos de ellos sean incompatibles.

**Cruce *Antienemies* (Fichero 2)**



**Cruce PMX (Fichero 2)**




En contraste con el fichero 1 en este caso al no existir alumnos incompatibles entre sí el resultado del método de cruce es notablemente peor que el cruce PMX y en general cualquier otra técnica de cruce.

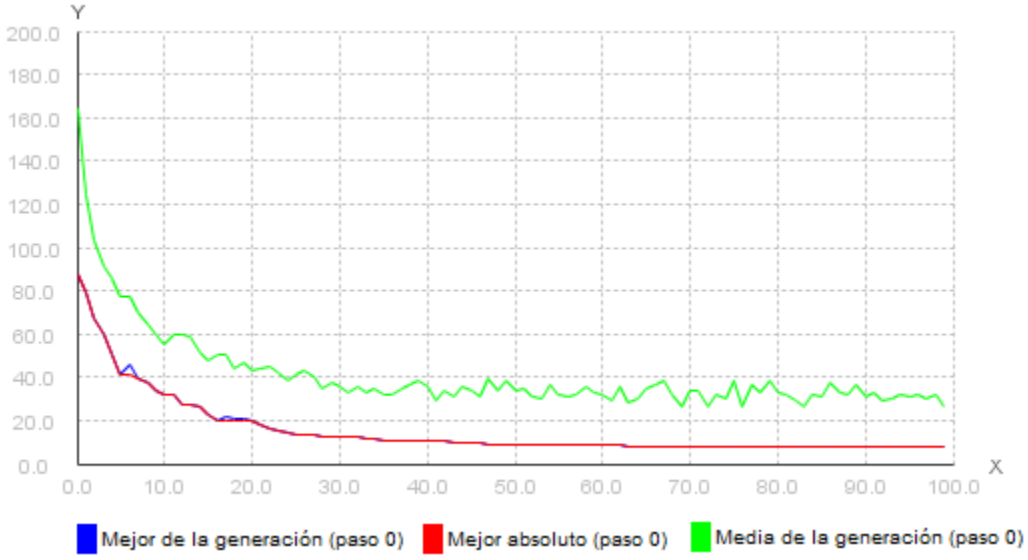
## 3.3. Ensayo 3. Mutación heurística

Dada la siguiente parametrización:

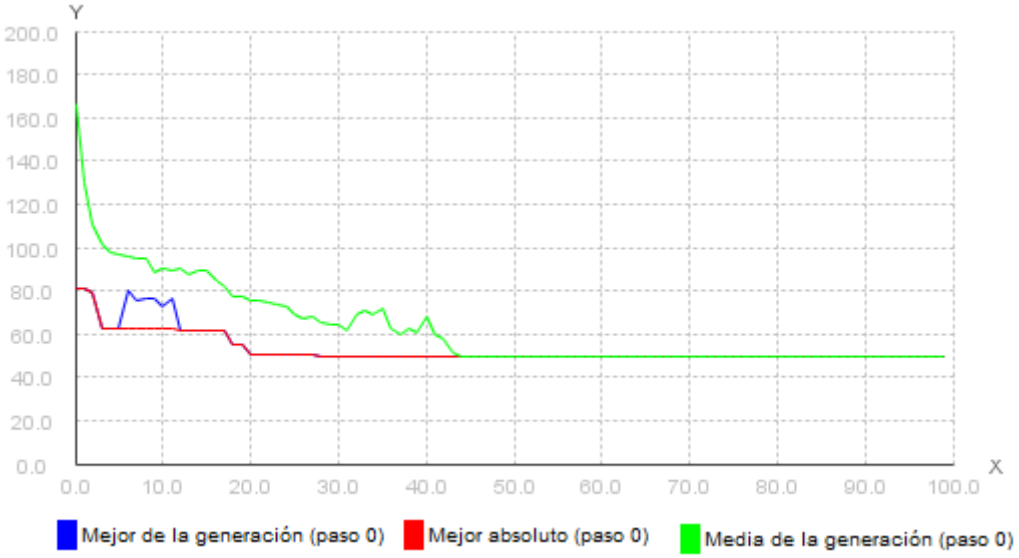
| Parámetro | Valor |
|---|---|
| Función de evaluación α | 0.5 |
| Tamaño de grupos m | 4 |

| Parámetro | Valor |
|---|---|
| Tam. Población | 100 |
| Prob. Cruce | 0.6 |
| Prob. Mutación | 1 |
| Número de generaciones | 100 |
| Selección | Ruleta |
| Cruce | OX |

Como cabe esperar a medida en esta mutación a medida que modifiquemos el parámetro del número de genes permutados se espera obtener un mejor resultado en la mutación de individuos, puesto que existen una mayor probabilidad de mejora, hasta el punto de que la población completa puede mutar. Llega un momento en el que la media de la población coincide con el mejor individuo, o como sino realizará mutación alguna.




**Mutación heurística con 5 genes seleccionados y permutados.**

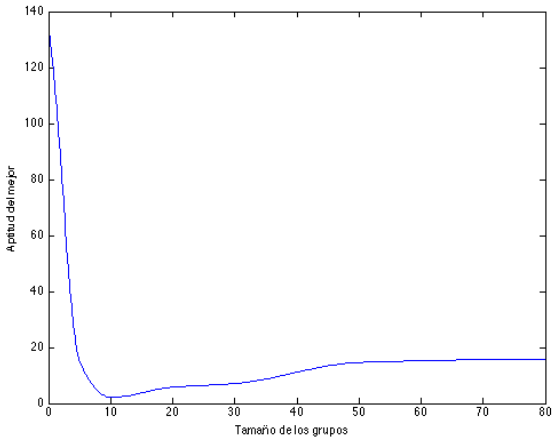



**Mutación heurística con 2 genes seleccionados y permutados.**

## 3.4. Ensayo 4. Función de evaluación

Sabemos que la función de evaluación actual tiene en cuenta tanto la similitud de las notas medias de los grupos como las incompatibilidades que se den en los grupos formados. Teniendo eso en cuenta, y valorando de la misma forma ambos factores ($\alpha = 0.5$) podemos observar:

| Tamaño del grupo | Aptitud del mejor (media de 5 ejecuciones) |
|---|---|
| 2 | 85.05 |
| 5 | 15.06 |
| 10 | 2.13 |
| 20 | 5.81 |
| 30 | 7.08 |
| 50 | 14.64 |
| 80 | 15.78 |




Vemos que aumentar el tamaño de grupo mejora los resultados, obtiene grupos más homogéneos. Sin embargo, a medida que aumenta el tamaño del grupo, también lo hace el número de incompatibilidades, por lo que, a pesar de que las medias de los grupos son idénticas, los resultados obtenidos empeoran.
Para este ensayo se utilizó ruleta como método de selección, PMX como operador de cruce e intercambio como operador de mutación.

## 4. Conclusiones

Podemos intuir rápidamente que el algoritmo puede confeccionar grupos más homogéneos de alumnos según aumentamos el tamaño de los mismos (si los grupos son demasiado pequeños, las posibilidades de compensarlos se ven limitadas), pero sólo hasta cierto punto. Pasado dicho punto, los resultados no son tan buenos, ya que a mayor tamaño del grupo, menor número de grupos se podrán hacer y, por tanto, mayor posibilidad hay de que en el mismo grupo acaben alumnos con afinidades opuestas (aumentando así el número de incompatibilidades).

Para finalizar, podemos concluir con que, en general, no se puede fijar un conjunto de valores para los parámetros que pueda ser considerada óptimo para todos los problemas, debemos estudiar las características de la función que se desee optimizar y estadistícamente determinar la configuración óptima que nos proporcione un mejor comportamiento del algoritmo para nuestro problema.